Testing Extended Visual Cryptography in Python

Diana Laura Aguilar Cervantes
School of Engineering and Sciences
Tecnológico de Monterrey, CEM
Atizapán de Zaragoza, Estado de México
Email: A01751168@itesm.mx

Constanza Marini Macouzet
School of Engineering and Sciences
Tecnológico de Monterrey, CEM
Atizapán de Zaragoza, Estado de México
Email: A01332485@itesm.mx

***Abstract*—Currently, a growing number of activities in a wide range of applications, involve sharing information on the internet. In this context, visual cryptography has attracted a considerable interest because of its crucial role in cybersecurity. In this paper, we test the techniques for visual cryptography proposed by Dhiman and Kasana by providing the results of their implementation in Python™, rather than in MatLab®. We tested two techniques: (3,3)-EVCT and (2,3)-EVCT. They both work with meaningful shares; however, they are different because (3,3)-EVCT requires the three generated shares to reconstruct the original image whereas (2,3)-EVCT can do it by using only two out of three shares. Our results show the effectiveness of these previously proposed techniques.**

1. Introduction

In recent years, sharing information through Internet has become a daily activity, and important operations, such as banking activities and electronic commerce, are also performed via Internet applications. Therefore, there is an urgent need to ensure information safety in this open network environment of exchanges. Traditional cryptographic technologies are commonly used to protect shared information; these techniques disorder the data during the encryption process and then it is recovered using the correct key. Moreover, the decryption can scarcely be done without the key [3].

Images are an extensively exchanged type of multimedia information, so there are techniques that allow sharing them in a secure and simple manner, mainly Visual Cryptography (VC). The pioneers of VC are Naor and Shamir who discovered how to decode confidential data without any computation [5]. The main advantage of traditional VC is decrypting secret images by human eye without any cryptographic computation by stacking different transparencies. It is pertinent to highlight that human eye can easily identify the content of a secret image even though the contrast is degraded by 50% [3]. The technique of (k, n)-VC, n shares of the secret image are generated by the encryption process. The recipient with k shares stacks them to reveal the original secret image; less than k shares would not recover the original secret image [2].

Several studies about visual cryptography have been published, however, most of them focus on black-and-white images [3]. However, Koga and Yamamotq proposed the mentioned technique (k, n)-VC for gray-scale and full colored images [2]. Nonetheless, (k, n)-VC technique shares resemble to random dot stereograms without visual meaning [4]. Therefore, it may be more evident that something is hidden in those shares. Ateniese et al., and Hwang and Chang concluded that a new gray-scale picture can be generated by directly superposing two meaningful pictures, called Extended Visual Cryptography Technique (EVCT) [1], [4]. Hou proposed three different (k, n)-VC methods based on halftoning technique and color decomposition in order to encode a full-color image [3]. Normally, VCT are measured by the level of the recovered image contrast and all the distinct methods results in a different contrast value. Contrast measures the difference between a black and a white pixel in the reconstructed image, hence in different recovered image sharpness [1].

For the purpose of understanding how the encryption and decryption of a color image are performed, some basic principles of color will be given. There are two models which describe the constitution of colors: the additive and subtractive models; one is the opposite of the other. In the additive model, the primary colors are red, green and blue (RGB), while in the subtractive model the primary colors are cyan, magenta and yellow (CMY). The desired colors are obtained by mixing different quantities of each primary color; here we will only focus on RGB color images because this model is used in monitors and electronic screens. Since the additive model is generated by sending color light, the more mixed colored lights are sent, the brighter the light will be. Therefore, when all the primary colors are mixed with equal intensity, the resulting light will be white. For this reason, any color mixed with white light results in white color, so the encryption blocks should be filled with red, green, blue and black colors. Moreover, in true color systems of 24-bits, each color is represented by 8 bits resulting in 256 variations of scale, which produces 16.77 million possible colors [3]; the first byte quantifies for the blue channel, the second is for green, and the third one is for red.

In this article, two different EVC approaches for true colored images, proposed by Dhiman and Kasana, are tested:

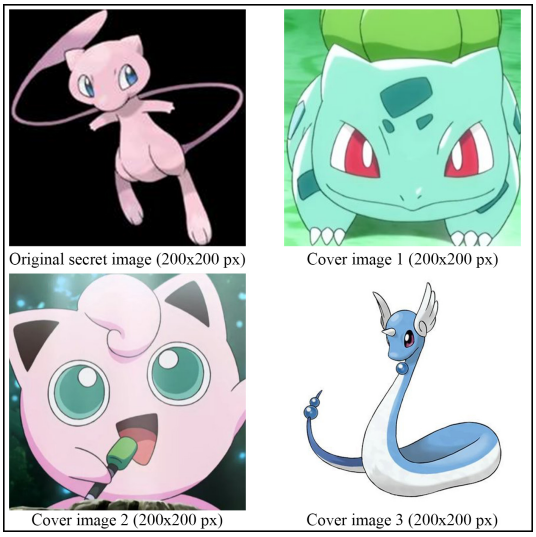

Figure 1. Tested images used in the evaluation.

(3, 3)-EVCT and (2,3)-EVCT. Both approaches store the RGB information of the original secret image in each share, which are covered by meaningful images. The recovered secret image is continuous in tone and without any loss of sharpness in each of the techniques. In (3, 3)-EVCT each share contains the information of one of each RGB component, so in order to reveal the secret image the three shares must be acquired. On the other hand, in (2, 3)-EVCT each share encloses the information of two of each RGB channel, i.e. RG, GB and RB, therefore only two shares are required to decrypt the original secret image [2]. The main purpose of this work is to construct the procedures of encryption and decryption of both methods in Python and compare the resulting quality of the recovered image, as well as the time taken by each approach

1.1. Methods

The tested techniques, (3, 3)-EVCT and (2, 3)-EVCT from [2] have a similar procedure, however, with a small difference. For (3, 3)-EVCT, first, through RGB color decomposition three shares are generated, one for each channel; in this step, the original secret image is expanded in size. Afterwards, each cover image is inserted into one of each shares in order for these shares to be meaningful; these shares can be send over an insecure communication line. For the decryption step the three shares must be acquired. The shares are stocked to get a matrix, which is used to recover the original secret image. The only difference between both techniques is that the shares of (2, 3)-EVCT contain the information of two channels, i.e. RG, GB, RB. As a result, the secret image can be decrypted by acquiring only two out of the three shares.

The formation of the original secret image shares is carried out by expanding each RGB color pixel into 5x5 array; each bit of the 8-bits color channel are arranged in the 5x5 array in such a way that each cell does not contain

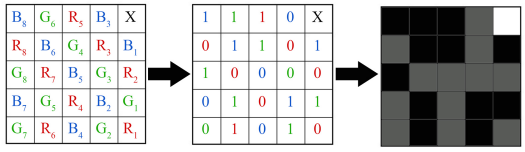

Figure 2. Example scheme of 24 bits allocated in 5x5 array.

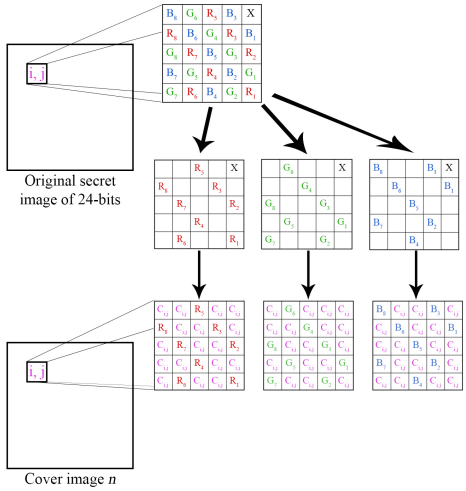

Figure 3. Encryption procedure of (3, 3)-EVTC.

the same color as its immediate vertical and horizontal neighbor. This procedure is explained in Figure 2. Moreover, the images used for the evaluation of the techniques are shown in Figure 1; notice that the cover images are of the same size as the original secret image.

1.1.1. Encrypting algorithms. The encrypting algorithm applied in (3, 3)-EVCT is presented in Figure 3. Subsequently, each step of this algorithm is clarified.

1) From a true colored RGB image of 24-bits, the original secret image, extract 8-bits of each color channel of the (i, j)th pixel.
2) Expand the (i, j)th pixel to a 5x5 array for each channel and fill it with the shares of each channel as presented in Figure 2. If a bit is 1, fill with black color at that position and if a bit is 0, fill it with dark gray. The 25th position must be left as it is.
3) Insert the cover images into each share. Fill the color of the (i, j)th pixel of each cover image into the empty position of each of the channel shares, i.e. cover image 1 into share R, cover image 2 into share G, and cover image 3 into share B.
4) Repeat steps 2 and 3 until all the pixels of the original secret image are processed and covered.

The following encrypting algorithm applied in (2, 3)-EVCT is illustrated in Figure 4.

1) From a true colored RGB image of 24-bits, the original secret image, extract 8-bits of each color channel of the (i, j)th pixel.

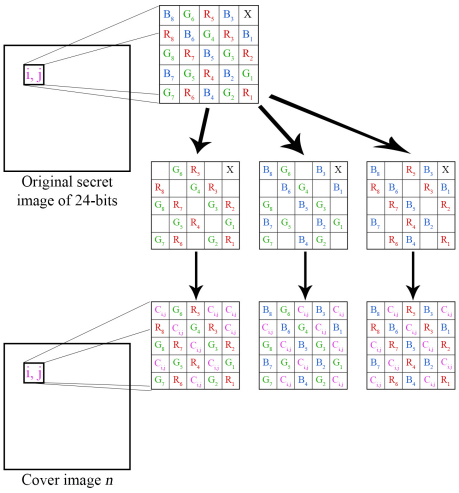

Figure 4. Encryption procedure of (2, 3)-EVCT.

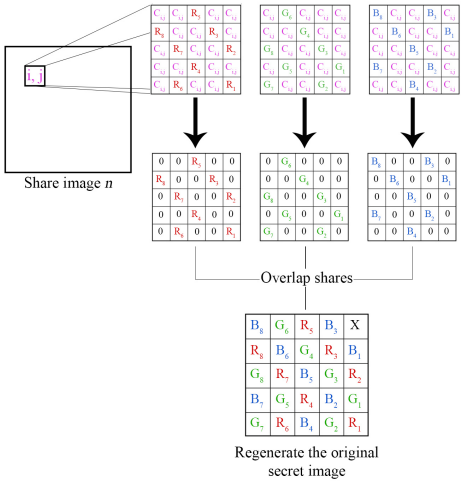

Figure 5. Decryption procedure for (3, 3)-EVCT.

2) Expand the (i, j)th pixel to a 5x5 array for RG, GB and RB channels.
3) Fill the array of share RG with the 8 bits of R component and 8 bits of G component; the array of share GB with 8 bits of G channel and 8 bits of B channel; and the array of share RB with 8 bits of R component and 8 bits of B component as presented in Figure 2. If a bit is 1, fill with black color at that position and if a bit is 0, fill it with dark gray. The 25th position must be left as it is.
4) Insert the cover images into each share. Fill the color of the (i, j)th pixel of each cover image into the empty position of each of the channel shares, i.e. cover image 1 into share RG, cover image 2 into share GB, and cover image 3 into share RB.
5) Repeat steps 2 to 4 until all the pixels of the original secret image are processed and covered.

1.1.2. Decrypting algorithms. The decrypting algorithm applied in (3, 3)-EVCT is presented in Figure 5.

1) Using the three meaningful shares, for each (i, j)th 5x5 array in each share, substitute the colored cells of the cover image with zeros.
2) Superimpose the converted shares by performing logical OR operation in order to obtain a matrix with information of the RGB components, in an array of 5x5, for each pixel of the original secret image.
3) Recover the secret image employing the generated matrix by attaining 8 bits of each RGB channel from the specific locations and unifying the 5x5 array into one color pixel.

The following decrypting algorithm employed in (2, 3)-EVCT is illustrated in Figure 6.

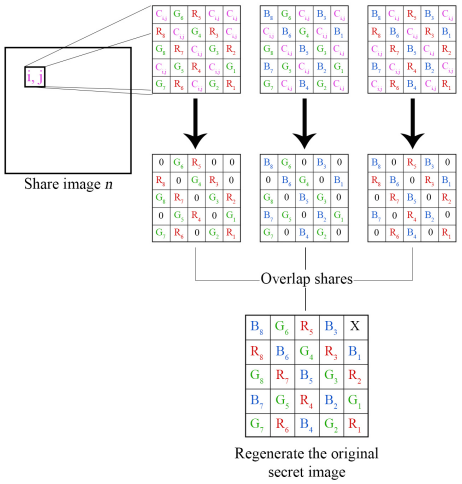

Figure 6. Decryption procedure for (2, 3)-EVCT.

1) Using two meaningful shares, for each (i, j)th 5x5 array in each share, substitute the colored cells of the cover image with zeros.
2) Superimpose the converted shares by performing logical OR operation in order to obtain a matrix with information of the RGB components, in an array of 5x5, for each pixel of the original secret image.
3) Recover the secret image employing the generated matrix by attaining 8 bits of each RGB channel from the specific locations and unifying the 5x5 array into one color pixel.

2. Results and discussion

The obtained results are presented in Figure 7 and Figure 8 for (3-3)-EVCT and (2-3)-EVCT, respectively. As shown

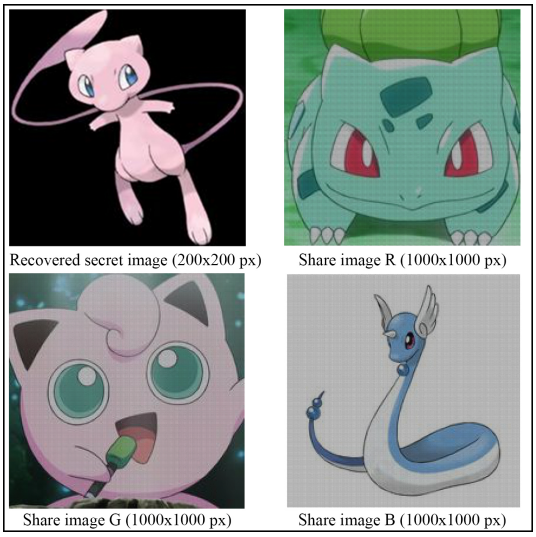

Figure 7. Shares and recovered secret image from (3-3)-EVCT.

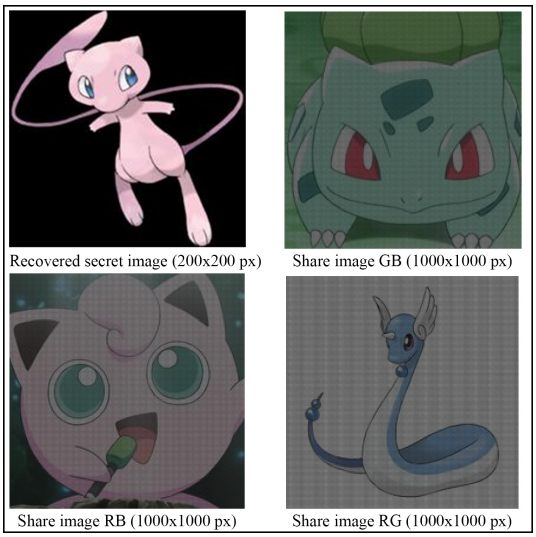

Figure 8. Shares and recovered secret image from (2-3)-EVCT.

in the figures, the recovered secret image has dimensions of 200x200, same as the original secret image. The shares of both techniques have dimensions of 1000x1000, due to the pixel expansion, and are meaningful.

The main advantage of VC is that secret images can be decrypted only by human eyes, without using any mathematical computation. However, EVCT decryption employs the superposing of meaningful images applying a simple logical operation, such as OR, AND or XOR. Many different (k-n)-EVCT techniques have been proposed for black and white, gray-scale, halftone and true color images. Nonetheless, the proposed techniques of Dhiman and Kasana surpasses others in resolution and contrast of the recovered secret image, time of execution, and number of colors of the secret image. These methods provide share images which are meaningful and cannot be decrypted without the information of the pixel expansion order. Also, they are fast and low resourceful methods taking less than 3 seconds for the whole process.

As previously mentioned, Dhiman and Kasana performed the encryption and decryption process in less than 3 seconds with a MATLAB program. Our tested procedure was done in approximately 1 minute with a Python script. The difference in time may be caused by the "for loops" employed for mapping each pixel of the original secret image, superposing the share cover on the original secret image, and decoding the secret image from the meaningful shares. In any case, the time taken to execute our procedure was less than other mentioned articles in [2]. Our recovered secret image has the same dimension, is of continuous color and with the same contrast as the original secret image.

In conclusion, by using these techniques, a true color image can be shared without any loss of information, hence dimensions, resolution and contrast. Moreover, the procedure of encryption and decryption are simple and uses few computational resources, i.e. memory and time. Additionally, images with high dimensions can be analyzed as well as medical images, geographic maps and satellite photographs are suitable to be shared securely. We encourage to perform these techniques with more than one secret image encrypted in meaningful shares.

References

[1] Giuseppe Ateniese, Carlo Blundo, Alfredo De Santis, and Douglas R Stinson. Extended capabilities for visual cryptography. *Theoretical Computer Science*, 250(1-2):143–161, 2001.

[2] Kirti Dhiman and Singara Singh Kasana. Extended visual cryptography techniques for true color images. *Computers & Electrical Engineering*, 70:647–658, 2018.

[3] Young-Chang Hou. Visual cryptography for color images. *Pattern recognition*, 36(7):1619–1629, 2003.

[4] Ren-Junn Hwang and Chin-Chen Chang. Hiding a picture in two pictures. *Optical Engineering*, 40, 2001.

[5] Moni Naor and Adi Shamir. Visual cryptography. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 1–12. Springer, 1994.